

IN THE CLAIMS:

Please amend the claims as follows:

1. (Canceled) A method of providing a user access to functional modules from within an application comprising:
 - collecting runtime metadata relating to the application;
 - obtaining a list of all functional modules accessible from within the application;
 - identifying a limited subset of the functional modules that will successfully execute, based on the runtime metadata and metadata associated with the functional modules; and
 - providing an interface presenting the user with the limited subset of functional modules that will successfully execute.
2. (Canceled) The method of claim 1, wherein identifying a limited subset of the functional modules that will successfully execute, based on the runtime metadata, comprises:
 - obtaining metadata associated with the functional modules; and
 - comparing the runtime metadata to the metadata associated with the functional modules.
3. (Canceled) The method of claim 2, wherein the metadata associated with at least one of the functional modules comprises at least one of: one or more input parameters required for successful execution of the functional module, one or more output parameters required for successful execution of the functional module, and a credential of a user authorized to execute the functional module.
4. (Canceled) The method of claim 1, wherein at least one of the functional modules is a plug-in component of the application.
5. (Canceled) The method of claim 1, wherein at least one of the functional modules is an external application.

6. (Canceled) The method of claim 1, wherein collecting runtime metadata comprises collecting session information during a logon procedure.
7. (Canceled) The method of claim 1, wherein collecting runtime metadata comprises collecting information regarding a current operating state of the application.
8. (Canceled) The method of claim 1, wherein identifying a limited subset of the functional modules that will successfully execute comprises determining if a resource required to execute a functional module is available.
9. (Canceled) The method of claim 8, wherein the resource is a server.
10. (Currently Amended) A method of providing a user access to functional modules from within an application, used to build queries, issue queries and/or, view query results, during a query session comprising:

assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database, wherein the assigned metadata requirements specify conditions required for successful execution of the functional module;

collecting runtime metadata relating to a query session, wherein the metadata is collected after the composition of a query;

obtaining a list of [[all]] functional modules that are accessible from within an application used during the query session;

identifying a limited subset of the functional modules in the list that will successfully execute, based on the runtime metadata and metadata associated with functional modules by comparing the collected runtime metadata with the assigned metadata requirements; and

providing an interface presenting the user with [[a]] the identified limited subset of functional modules that will successfully execute.
11. (Original) The method of claim 10, wherein the runtime metadata comprises attributes of fields involved in a query or query results.

12. (Original) The method of claim 10, wherein the runtime metadata comprises content contained in query results.

13. (Canceled) The method of claim 10, wherein identifying a limited subset of the functional modules that will successfully execute, based on the runtime metadata, comprises:

obtaining metadata associated with the functional modules; and

comparing the runtime metadata to the metadata associated with the functional modules.

14. (Currently Amended) The method of claim [[13]] 10, wherein obtaining metadata associated with the functional module comprises examining a signature validation.

15. (Currently Amended) The method of claim [[13]] 10, wherein the metadata associated with at least one of the functional modules comprises at least one of: one or more input parameters required for successful execution of the functional module, one or more output parameters required for successful execution of the functional module, and a security credential required to execute the functional module.

16. (Original) The method of claim 10, wherein at least one of the functional modules analyzes query results.

17. (Original) The method of claim 16, wherein:

the runtime metadata comprises the names of fields in a result set; and

the limited subset of functional modules comprises functional modules requiring data from fields in the result set as inputs.

18. (Original) The method of claim 10, wherein the runtime metadata comprises information related to a query building session.

19. (Original) The method of claim 18, wherein:

the information related to the query building session comprises a specified focus of the query; and

identifying a limited subset of the functional modules that will successfully execute comprises identifying functional modules associated with the specified focus.

20. (Currently Amended) A computer readable medium containing a program which, when executed, performs operations for providing a user access to functional modules from within an application, comprising:

assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database, wherein the assigned metadata requirements specify conditions required for successful execution of the functional module;

collecting runtime metadata relating to a query session the application;

obtaining a list of [[all]] functional modules accessible from within the application

identifying a limited subset of functional modules that will successfully execute, ~~based on the runtime metadata and metadata associated with functional modules by comparing the collected runtime metadata with the assigned metadata requirements;~~ and

providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute.

21. (Original) The computer readable medium of claim 20 wherein the application is a query building application.

22. (Canceled) A system for providing a user access to functional modules comprising:

a plurality of functional modules, each having associated metadata;

an application from which the functional modules are accessible, wherein the application is configured to collect metadata and present to a user a limited subset of the functional modules, based on the runtime metadata and the metadata associated with the functional modules.

23. (Canceled) The system of claim 22, wherein at least one of the functional modules is a plug-in component of the application.
24. (Canceled) The system of claim 22, wherein at least one of the functional modules is an external application.
25. (Canceled) The system of claim 22, wherein the runtime metadata comprises a user's session information.
26. (Canceled) The system of claim 22, wherein the runtime metadata comprises a system state information.
27. (Currently Amended) A data processing system for providing a user access to functional modules from within an application comprising:
a data repository;
a plurality of functional modules, each having associated metadata requirements that specify conditions required for successful execution of the functional modules;
an application from which the functional modules are accessible, wherein the application is configured to collect runtime metadata after the composition of a query and present to a user a limited subset of the functional modules that will successfully execute, as determined by the application based on the collected runtime metadata and the metadata requirements associated with the functional modules.
28. (Original) The data processing system of claim 27, wherein the data repository comprises XML data structures used to store runtime metadata.
29. (Original) The data processing system of claim 27, wherein the data repository comprises relational database tables used to store runtime metadata.